

Modélisation et spécification – Master 2 Informatique

TD 2 : Systèmes de transitions étiquetés

Tampon avec acquittement

Dans ces exercices on vous demande de modéliser un tampon à une place qui, pour chaque donnée reçue, envoie un acquittement à l'émetteur. Plus précisément, en plus des actions `get` et `put`, ce tampon comporte :

- une action `put-ack` sur lequel il reçoit l'acquittement et
- une action `get-ack` sur lequel il envoie l'acquittement reçu.

Exercice 1 :

Premier essai

Une implémentation possible de ce type de tampon reçoit et envoie la donnée, puis reçoit et envoie l'acquittement.

1. Donner une modélisation de cette implémentation.
2. On compose en parallèle 2 copies du modèle obtenu tel que les ports `get` et `put-ack` communiquent avec les ports `put` respectivement `get-ack` du voisin de droite. Pensez-vous que le modèle obtenu corresponde à un tampon à deux places avec acquittement ?

Exercice 2 :

Deuxième essai

Une autre implémentation du tampon avec acquittement envoie l'acquittement sur `get-ack` immédiatement après avoir reçu la donnée sur le port `put`, ensuite il renvoie la donnée sur `get` et attend son acquittement sur `put-ack`.

1. Donner une modélisation de cette implémentation.
2. On compose en parallèle 3 copies du modèle obtenu selon les indications de l'exercice précédent. Qu'en pensez-vous par rapport au modèle de l'exercice précédent ?

Un autre exemple

Exercice 3 :

Le passage à niveau

On considère un système de passage à niveau de voies ferrées avec trois entités :

1. Le train qui en boucle est soit proche, soit sur le passage à niveau soit loin
2. Un contrôleur qui attend qu'un train approche, puis lève la barrière puis attend qu'un train s'éloigne et baisse la barrière
3. Une barrière qui peut être soit levée, soit baissée

Proposez une modélisation des trois entités sous forme de STE et donnez ensuite une définition cohérente pour le STE du système global.

Encore un exemple

Exercice 4 :

Manipulation de verrous

On considère un système fait de deux processus $P1$ et $P2$ et deux verrous $V1$ et $V2$ de type mutex.

Chaque verrou a deux états, soit pris, soit libre.

Le processus $P1$ réalise au sein d'une boucle infinie répétitive les actions suivantes : il teste si le verrou $V1$ est libre, si c'est le cas, il le prend, puis il prend le verrou $V2$, il réalise alors sa tâche et ensuite il libère $V2$ puis $V1$; si en revanche le verrou $V1$ n'est pas libre, alors il prend le verrou $V2$ puis le verrou $V1$, il réalise sa tâche puis il libère $V2$ et $V1$. (On rappelle que lorsque l'on dit que $P1$ prend le verrou

$V2$ cela veut dire qu'il le prend si il est libre, et que sinon il attend que le verrou soit libre et lui soit donné).

Le processus $P2$ réalise quant à lui au sein d'une boucle infinie répétitive les actions suivantes : il prend le verrou $V2$ puis il prend le verrou $V1$, il réalise sa tâche et ensuite il libère $V1$ puis $V2$.

On souhaite modéliser ce système par un système de transitions étiquetées. Pour cela on considèrera l'alphabet suivant pour les actions :

- $S-i$ (pour i valant 1 et 2) : prend le verrou V_i
- $V-i$ (pour i valant 1 et 2) : libère le verrou V_i
- $L-i$ (pour i valant 1 et 2) : teste positif si le verrou V_i est libre
- $P-i$ (pour i valant 1 et 2) : teste positif si le verrou V_i est pris
- $T-i$ (pour i valant 1 et 2) : P_i fait sa tâche

On supposera de plus que dans l'état initial les deux verrous sont libres.

1. Donner une modélisation sous forme de système de transitions étiquetées du verrou $V1$.
2. Dédire une modélisation du verrou $V2$.
3. Donner une modélisation du processus $P1$.
4. Donner une modélisation du processus $P2$.
5. En déduire une modélisation pour tout le système.
6. Existe-t-il une trace dans le système obtenu amenant à un état sans successeur (c'est-à-dire un *deadlock*)?